LaTeX 排版心得

李东风

2013 年 12 月 12 日

目录

第一章 LaTeX 入门 **1**

1.1 LaTeX 基本知识 1

1.1.1 LaTeX 历史和特点 1

1.2 安装与设置 2

1.3 编辑和编译 2

1.3.1 在 TeXWorks 中编写 LaTeX 文件 3

1.3.2 在 WinEdt 中编写 LaTeX 文件 4

1.3.3 用命令行编译 5

1.3.4 编译为 HTML 5

1.4 LaTeX 源文件 5

1.4.1 LaTeX 输入文件 5

1.4.2 特殊字符 5

1.4.3 LaTeX 命令 6

1.4.4 输入文件的结构 7

1.5 CTeX LaTeX 源文件示例 8

1.6 一般文本排版 9

1.6.1 段落 9

1.6.2 标题、章节 9

1.6.3 环境 10

1.6.4 制表环境 `tabular` 10

1.6.5 浮动环境 10

1.7 数学公式入门 10

第二章 Emacs+AucTeX **15**

2.1 安装 15

2.2 Emacs 的基本编辑功能 17

2.3 用 Emacs+AucTex 进行 LaTeX 编辑编译 18

2.3.1 输入辅助 19
2.3.2 注释 19
2.3.3 数学符号输入辅助 19
2.3.4 大文件辅助 20
2.4 用 reftex 作引用 20

第三章 LaTeX 提高 **21**
3.1 编译结果 21
3.2 带书签的 PDF 文件 22
3.3 版式设置 22
3.4 大文件处理 23
3.5 插入图形 24
3.6 源程序排版 26

第四章 幻灯片演示制作 **29**
4.1 用 Beamer 制作幻灯片 29
4.1.1 基本的 beamer 幻灯片 29
4.1.2 编译速度 32
4.1.3 同时生成幻灯片和打印版本 32
4.2 用 PDFSlide 制作幻灯片 33
4.2.1 PPower4 软件 34

第五章 其他技巧 **36**
5.1 笔记 36
5.1.1 一些自定义命令 36
5.2 公式和符号问题 37
5.2.1 特定问题 37
5.3 其他问题 38
5.3.1 关于英文系统和中文系统的兼容性问题 38
5.3.2 一些小的问题 38

第一章　LaTeX 入门

本文先简单介绍 TeX 和 LaTeX 系统，然后是一些经验总结，其中包括作者用 LaTeX 和 PDFLaTeX 打印《统计软件教程》一书的心得，编辑一些数学书籍的经验，和使用 LaTeX 制作课堂演示课件的一些经验。新的 LaTeX 项目应该使用 XeLaTeX+UTF-8 的方法，这样生成 PDF 比较简单，而且可以用到系统安装的所有字体。

本文排版结果展示了一些特殊技巧，要知道自己如何使用这些技巧要参考本文的 LaTeX 源程序。

1.1　LaTeX 基本知识

1.1.1　LaTeX 历史和特点

TeX 是著名计算机专家 Donald E. Knuth 发明的一个用于文章排版的计算机软件系统，也是一种编程语言。1982 年公开。以软件稳定性高著称。

LaTeX 是在 TeX 基础上是排版更方便的一个子系统。

LaTeX 和 MS Word 比较有如下特点:

- MS Word 是桌面文档系统，不能作为高质量印刷。强调 WYSIWYG。
- LaTeX 不提倡作者自己规定文章样式，而使用统一样式。强调文章的逻辑结构而不是直观样式。
- 很多杂志投稿接受 LaTeX 电子版会提供自己的模板，只要使用他们的模板就可以完美地融入整个杂志中。
- 强项在于数学公式和特殊符号丰富易用。
- 容易生成公式图表编号、引用、交叉引用、目录。
- 用 LaTeX 可以容易地管理大型文档。
- LaTeX 可以输出成 Postscript、PDF、HTML 等格式。

- 一些数学计算软件可以直接生成 LaTeX 表格。
- TeX 和 LaTeX 是免费的! 各操作系统都支持。支持中文。

1.2 安装与设置

LaTeX 有许多可选的安装方式，功能各有特点。为了方便处理中文，在 MS Windows 系统中作者建议使用 CTEX 安装包。尽管有其他方法也能处理中文，但现在比较完善的还是 CTEX。

从 `www.ctex.org`中获取软件。最新版本是 2.8 版。安装完成后，可以使用同时安装的 TeXWorks 输入源文件和编译、查看。为了能够在 TeXWorks 中能使用 GBK 编码的中文，选菜单“编辑 —选项”，在对话框的“编辑器” 页面选编码为“System”。

新的项目应该使用 XeLaTeX + UTF-8 的办法，选菜单“编辑 —选项”，在对话框的“编辑器” 页面选编码为 “UTF-8”。选菜单“编辑 —选项”，在对话框的 “排版” 页面选处理工具 “默认” 为 “XeLaTeX”。含有中文的输入的每个 LaTeX 文件前两行都应为

```
% !TEX encoding = utf8
% !TEX program = xelatex
```

Emacs 编辑器是在 Linux 及其他 Unix 类操作系统上广泛使用的一个功能极为强大的集成工作环境，有 MS Windows 版本。在安装了 AucTeX 包后可以作为一个集成的 LaTeX 编辑、编译、浏览、打印系统 (需要 MikTeX 作为后台支持)。在编辑 LaTeX 文件时有很好的语法彩色加亮功能，支持源文件位置到 DVI 位置的正、反向对应, 还可以预览文内的公式和图形。

注意：Emacs 的使用和其他编辑器差别较大，可能需要较长的适应过程。新手建议使用 TeXWorks。

为了安装设置 Emacs+AucTex+CTEX，请参见第二章。

1.3 编辑和编译

LaTeX 源文件需要经过编译才能查看和打印。一般编译为 DVI、PDF、Postscript、HTML 等格式。

DVI 格式是一种编译结果，用专门的查看程序 YAP 查看，可以在写作中间过程中使用。为了提供给大众浏览还是应该变成 PS 或 PDF 格式。可以把 DVI 转换为 PS 和 PDF，也可以直接把 LaTeX 文件用 PDFLaTeX 或 XeLaTeX 编译为 PDF，XeLaTeX 的源文件应该用 UTF-8 编码保存。

编译过程会生成一些辅助文件，在得到结果后可以删除这些辅助文件。辅助文件包括 *.log, *.aux, *.toc, *.out, *.nav, *.vrb 等文件。

1.3.1 在 TeXWorks 中编写 LaTeX 文件

TeXWorks 是一个便于初学者使用而又具有强大功能的 LaTeX 编辑、编译、查看集成环境, 是免费的。编译结果为 PDF。它支持 LaTeX 的语法加亮显示，源程序和编译 PDF 结果左右并排放置，可以从源程序跳到结果中对应位置，也可以从 PDF 结果跳到源程序相应位置。可以通过 LaTeX+CJK 支持中文，也可以用 XeTeX+UTF-8 直接使用 UTF-8 编码的中文。

TeXWorks 提供了几个简单模板，初学者可以用这些模板作为自己的开始样式。用户自己的模板也可以放在 `C:\Documents and Settings\`你的用户名`\TeXworks` 下面的 templates 目录中。

为了使用 GBK 编码的 CJK 中文，可以在“编辑 —选项 —编辑器 —编码”中选择 System，这样缺省使用系统的中文编码，也可以在文件首行写特殊注释

```
% !TEX encoding = System
```

新的项目应该使用 XeLaTeX + UTF-8 的办法，选菜单“编辑 —选项”，在对话框的 “编辑器” 页面选编码为 “UTF-8”。选菜单“编辑 —选项”，在对话框的 “排版” 页面选处理工具 “默认” 为 “XeLaTeX”。含有中文的输入的每个 LaTeX 文件前两行都应为

```
% !TEX encoding = utf8
% !TEX program = xelatex
```

为了编译，只要按图标栏的“排版”菜单即可。用 Ctrl+Click 或右键菜单可以在源程序和对应的排版结果之间来回跳转。注意，在使用 beamer 包时，`\maketitle` 命令会导致逆向查找功能失败。

编译输出显示在编辑区和状态行之间，如果编译出错，会在输出和状态行之间打开一个控制台行，在此回车可继续，输入 x 后回车可以停止编译。

编译会产生一些 *.aux, *.log, *.out 等辅助文件，可以用“文件 —删除辅助文件”菜单来删除。

要注释掉选定的行，用菜单“格式 —注释掉”。命令“格式 —取消注释”可以去掉注释。

TeXWorks 有辅助输入功能。打入一个命令的一部分然后按 TAB 键可以自动完成这个命令，比如打入 latex 然后按 TAB 键可以变成LaTeX。打入以 b 为前缀环境名或部分环境名可以插入环境，如果有多个环境可以匹配可以用 TAB 键切换。插入的环境中有圆点用来定位到某输入位置，用 Ctrl+TAB 键可以切换这些输入位置。

如果编辑窗的字太小或字体不合适，可以用“格式 t—字体...”菜单修改。

在编写英文文件时，可以选菜单设置“编辑 —拼写 —en_US”来打开拼写检查功能。

当文件比较长的时候，可以选菜单“窗口 —显示 —标签”显示文件的树状结构导航。

用“编辑 —选项...”菜单的“排版”页可以设置默认的编译命令，如 PDFLaTeX 或 XeLaTeX。从图标栏的“排版方式”下拉菜单可以临时改用某编译命令。如果某个文件总是用某一命令编译，可以在文件开头加上特殊注释如

```
% !TeX program = xelatex
```

这指定此文件用 xelatex 编译。

如果使用主控文件和子文件，在子文件的开头应加上注释行

```
% !TeX root = mypap.tex
```

其中 mypap.tex 是主控文件名。

1.3.2 在 WinEdt 中编写 LaTeX 文件

WinEdt 和是一个一般性的文本文件编辑器，是共享软件，提供了对 LaTeX 编辑、编译的较好支持。支持 LaTeX 的语法高亮显示，辅助输入，编译和查看结果的快捷方式。可以编译为 DVI 或 PDF 格式。

WinEdt 中有编译用的快捷图标, 可以用 LaTeX 把文件编译为 DVI 格式，用 XeLaTeX 把文件编译为 DVI 格式，用 PDFLaTeX 把文件编译为 PDF 格式，可以把 DVI 格式转换为 Postscript 格式 (*.ps) 或 PDF 格式，把 Postscript 格式转换为 PDF 格式。可以查看编译的 DVI 或 PS、PDF 文件。在查看 DVI 时双击某处可以反向查找源程序中对应位置 (TeXWorks 和 Emacs+AucTeX 也具有这样的功能)。

1.3.3 用命令行编译

如果设置了程序路径，也可以在一个命令行窗口进行编译, 如:

```
latex examp-eng
yap examp-eng
dvips examp-eng
dvipdfmx examp-eng
```

1.3.4 编译为 HTML

为了能够把公式显示在 HTML 结果中，需要预先安装好 ImageMagic 软件，确保路径中能找到。TeX2ht 软件可以把 LaTeX 源程序编译为 HTML 格式，用 htlatex 命令编译。如在命令行运行

```
htlatex latex02
```

应该预先把待编译的文件和图形放在一个单独的目录中。编译后可能会生成一些图形文件，这些图形文件要伴随生成的 HTML 文件。

1.4 LaTeX 源文件

1.4.1 LaTeX 输入文件

LaTeX 输入文件是一种计算机源程序，但是简单任务只需要很少的命令。多个空格和一个空格不区分，所以不能像 Word 那样用增加空格对齐。空行代表不同段落。

图1.1是最简单的一个输入文件 `examp-eng.tex`，可以用 latex、pdflatex、xelatex 编译。图1.2是最简单的一个 UTF-8 编码中文输入文件 `examp-ctex-utf8.tex`，需要用 XeLaTeX 编译。图1.3是最简单的一个 GBK 编码中文输入文件 `examp-ctex-gbk.tex`，可以用 latex 或 pdflatex 编译。

1.4.2 特殊字符

LaTeX 命令中要用到一些特殊字符，所以实际想印出这些特殊字符时必须使用替代命令。

\# \$ \% \^ \& _ \{ \} \~ \\.

```
\# \$ \% \^{} \& \_ $\{$ $\}$ \~{} $\backslash$.
```

```
\documentclass{article}
\begin{document}

Paragraph 1.
The input for \LaTeX is a plain
{\LARGE ASCII} text file. You can create it with any
text editor. It contains
the text of the document, as well as the commands
that tell LATEX how to typeset the text.

Paragraph 2.
``Whitespace'' characters, such as blank or tab,
are treated uniformly as
``space'' by LATEX. Several consecutive whitespace
characters are treated
as one ``space.'' Whitespace
 at the start of a line is generally ignored, and a
single line break is treated as ``whitespace.''.

\end{document}
```

图 1.1: 最简单的英文 LaTeX 输入文件

双引号左边用两个反单引号，右边用两个单引号。

连字符用 −− 代替: my–file。

破折号用 − − − 代替: Hero—who?。

1.4.3 LaTeX 命令

命令以反斜杠开头，后面英文字母组成的命令。如 \LaTeX。命令后面的空格被忽略。

命令可以有参数，写在大括号内，如 \texttt{name} 显示为 `name`。

为了限制一个命令的作用范围，把希望有作用的范围包在 {···} 内。如 {\Large 大字体 } 结果为大字体。

% 后面是注释。注释中的空格、换行不被解释，所以一个长单词如果用注释在结尾拆开最后结果是相连的。

```
% !TEX encoding = utf8
% !TEX program = xelatex

\documentclass{ctexart}

% 设置中文字体
\setCJKmainfont[BoldFont={黑体},ItalicFont={楷体}]{新宋体}

\begin{document}

LaTeX 不提倡作者自己规定文章样式，而使用统一样式。
强调文章的逻辑结构而不是直观样式。
很多杂志投稿接受 LaTeX 电子版会提供自己的模板，
只要使用他们的模板就可以完美地融入整个杂志中。
强项在于数学公式和特殊符号丰富易用。
容易生成公式图表编号、引用、交叉引用、目录。

``用 LaTeX 可以容易地管理大型文档。''
LaTeX 可以输出成 Postscript、PDF、HTML 等格式。
一些数学计算软件可以直接生成 LaTeX 表格。 TeX 和 LaTeX 是免费的!
各操作系统都支持。支持中文。



\end{document}
```

图 1.2: 最简单的中文 CTeX UTF-8 输入文件

1.4.4 输入文件的结构

\documentclass 命令规定使用的文档模板，是必须的。常用的有article、book、report、beamer等。

中文文档模板可用 ctexart、ctexbook、ctexrep等。

文档内容写在\begin{document} 和\end{document} 之间。\documentclass 和\begin{document} 之间的部分叫做序言 (preamble)，主要是引入一些额外的宏包 (macro package) 和作一些用户定制。引入宏包如\usepackage{CJK}。这是支持 GBK 编码的中文输入文件排版的

```
% !TEX encoding = GBK
% !TEX program =  pdflatex

\documentclass{ctexart}

\begin{document}

\LaTeX 不提倡作者自己规定文章样式，而使用统一样式。
强调文章的逻辑结构而不是直观样式。
很多杂志投稿接受 \LaTeX 电子版会提供自己的模板，
只要使用他们的模板就可以完美地融入整个杂志中。
强项在于数学公式和特殊符号丰富易用。
容易生成公式图表编号、引用、交叉引用、目录。

``用 \LaTeX 可以容易地管理大型文档。''
\LaTeX 可以输出成 Postscript、PDF、\, HTML 等格式。
一些数学计算软件可以直接生成 \LaTeX 表格。
\TeX 和 \LaTeX 是免费的!
各操作系统都支持。支持中文。

\end{document}
```

图 1.3: 最简单的中文 CTeX GBK 输入文件

宏包。

另外常用的宏包有: ctex—支持中文习惯的标题、目录等。graphicx—插图用。amsmath— 一些方便的公式、符号命令。

1.5 CTeX LaTeX 源文件示例

下面是一个基本的使用 CTeX 的框架 (用 GBK 编码保存):

```
\documentclass[a4paper]{ctexart}
\begin{document}

\include{mypap-inc}

\end{document}
```

其中的内容部分写在一个单独的文件 mypap-inc.tex 中，用 `\include{mypap-inc}` 可以把文件内容整体插入。这样拆成主文件、子文件的好处是可以写几个主文件，比如一个主文件用于生成 DVI，编译速度快；一个主文件用于生成 PDF，编译较慢但是结果交流容易；一个主文件用于生成课堂演示幻灯片；一个主文件用于 CCT，等等。这些主文件共享一个内容子文件。因为生成 DVI 和 PDF、幻灯片、使用 CCT 等所需的文件头不同 (采用的宏包不同，插图方式不同）所以用几个主文件来调用相同的内容文件是比较好的办法。如果使用一般的模板如幻灯类 beamer，可以载入 ctex 宏包。

要注意的是使用`\include` 命令会在插入内容的前后换页。

LaTeX 习惯在章节开始段首不缩进，为了使得开始段也段首缩进可以引入 indentfirst 宏包。

1.6 一般文本排版

1.6.1 段落

LaTeX 只需规定文章的结构: 段落、小节、节、章、标题、作者、日期等。

用空行分开的部分为段落。源文件中的段落内换行不影响最后的排版结果。一定不要在无意识中新开一个段落。比如，公式后面如果有空行则为新段落。

段落由句子组成，LaTeX 试图根据标点分开句子。

LaTeX 可以在行尾拆分单词以使得行内的空白宽度适中。

用 \\ 强制在段内换行。用 \newpage 强制换页。如果某些行难以拆分，会造成长行，这时可以在前面用一个 \sloppy 命令，可以在较前面拆分，造成较多行内空白，过后用 \fussy 命令恢复。可以用 \- 加入一个单词中提示可以在该位置换行。

1.6.2 标题、章节

用 \title 命令指定标题，\author 命令指定作者，\date 命令指定日期 (见 `latex04.tex`)。\maketitle 制作标题。

在 `article`文档内，用 \section{$\cdots$}, \subsection{$\cdots$}, \subsubsection{$\cdots$}, \paragraph{$\cdots$}, \subparagraph{$\cdots$} 指定文档结构。

在 `article`文档内还可以用 \part{$\cdots$} 命令把文档分成几个部分，

各部分章节仍统一编排。在 `book`文档内有更高一级的 \chapter{···}。用 \appendix 命令指定后面章节为附录，编号变成 A, B, C 等等。用 \tableofcontents 命令制作目录。

用 \emph{···} 强调内容。用 \footnote{···} 作脚注。

在某个章节用 \label{sec:intro} 之类作标签，可以在其它地方用 \ref{sec:intro} 引用这个章节。可以在其它地方用 \pageref{sec:intro} 引用该处的页码。

1.6.3 环境

用 \begin{ 环境名 } 开始，用 \end{ 环境名 } 结尾。中间包含允许的内容。

如: 列表 `itemize`环境。用 \item 命令开始一个条目。

如: 有序列表 `enumerate`环境。

如：居中环境 `center`环境。

原样输出环境 `verbatim`，行内可以用 \|···| 命令。

1.6.4 制表环境 `tabular`

开始环境有参数 \begin{tabular} {···}，参数如 `ccc`，表示有三列，都居中对其。`l`表示左对齐，`r`表示右对齐。`|`表示增加一条竖线。表格行用 \\ 结尾; 用 & 分开表格单元。表格行之间用 \hline 画线。

1.6.5 浮动环境

长表格占地多，可能无法在本地排开只好到下一页，造成前页多余空白。LaTeX 把表格包在 `table`环境中，则表格可以自动排到合适的位置。在 `table`环境中，用 \label{tab:person} 之类定义表格标签，然后在其它地方可以用 \ref{tab:person} 引用表格号码。

图形用 `figure`浮动环境。

1.7 数学公式入门

数学公式是 LaTeX 的强项。数学中所有符号基本都可以在 LaTeX 中找到。(参见: `symbols-a4.pdf`.)

行内公式包在 $ 和 $ 之间，或包在 \(和 \) 之间。如

```
行内公式 \(\int_0^1 \sum_{i=1}^n f(x_i, \theta) d\theta\)。
$a^{(2)}$ 正确。$a^(2)$ 错。分式 $\frac{a}{b}$。
```

显示结果为：
行内公式$\int_0^1 \sum_{i=1}^n f(x_i, \theta) d\theta$。$a^{(2)}$ 正确。$a^(2)$ 错。分式 $\frac{a}{b}$。

独立公式 (displayed formula) 包在 $$ 和 $$ 之间，或包在 \[和 \] 之间。 比如

```
$$
\int_0^1 \sum_{i=1}^n f(x_i, \theta) d\theta
$$
或
\[
\mbox{rain}^2 + \mbox{wind}^2 = c^2.  \tag{a'}
\]
\[
rain^2 + wind^2 = c^2.
\]
```

显示结果为

$$\int_0^1 \sum_{i=1}^n f(x_i, \theta) d\theta$$

或

$$\mathrm{rain}^2 + \mathrm{wind}^2 = c^2. \tag{a'}$$

$$rain^2 + wind^2 = c^2.$$

注意独立公式前后不要留有空行, 除非另起一段。行内公式和独立公式风格有所不同。

数学公式中的多个空格只能作为一个，不能空行，每个字母都认为是一个变量，需要一些普通文字可以用 \mbox{$\cdots$} 命令。

一些基本公式元素:

- ^ 后为上标，如 `a^2` 结果为 a^2, `a^{(2}}` 结果为 $a^{(2)}$。
- _ 后为下标。如 `a_1` 结果为 a_1。
- `a_1^2` 结果为 a_1^2。注意含多个字符的上下标需要用大括号包围，如 $e^{ij\lambda}$ 应输入为`$e^{ij\lambda}$`。

- 分式: \frac{ 分子 } { 分母 } 格式。如 $\frac{a}{b}$。
- 求和: \sum。如 $\sum_{i=1}^{n} \frac{1}{i^2}$。
- 积分: \int。如 $\int_0^1 f(x)dx$。

用 `equation`环境或 `align`环境排版自动编号的公式。(`align`环境需要 amsmath 宏包)。在环境内用如 \label{eq:abc1} 之类定义公式标签，在别处用 (\ref{eq:abc1}) 来引用公式号。

用 equation 环境排版编号公式的例子:

```
\begin{equation}
  a + b = c \label{eq:abc1}
\end{equation}
```

结果如

$$a + b = c \tag{1.1}$$

引用如公式 (1.1) 或(1.1)。

用 align 环境排版编号公式的例子:

```
\begin{align}
  a + b = c \label{eq:abc12}
\end{align}
```

结果如

$$a + b = c \tag{1.2}$$

用 `eqnarray`环境 `align`环境排版自动编号的多行公式，不同行用 \\ 分隔，`eqnarray`环境的公式把中间的等号改成 & = & 则可以对齐等号；`align`环境的公式把中间的等号改成 = & 则可以对齐等号。没有等号的行则在需要与其它行等号对齐的位置加上 &&(eqnarray 环境) 或 &(align 环境)。`eqnarray`环境和 `align`环境改成 `eqnarray*`环境和 `align*`则不编号，但可以用 \tag 命令手工编号。

用 eqnarray 环境排版编号多行公式的例子：

```
\begin{eqnarray}
  a + b &=& c \label{eq:abc21} \\
  a^2 + b^2 &=& c^2 \label{eq:abc22}
\end{eqnarray}
```

结果如：

$$a + b = c \tag{1.3}$$
$$a^2 + b^2 = c^2 \tag{1.4}$$

用 align 环境排版编号多行公式的例子:

```
\begin{eqnarray}
  a + b =& c \label{eq:abc31} \\
  a^2 + b^2 =& c^2 \label{eq:abc32}
\end{eqnarray}
```

结果如：

$$a + b = c \tag{1.5}$$
$$a^2 + b^2 = c^2 \tag{1.6}$$

用 align 环境排版编号的多行推导，用 \nonumber 命令取消某行的编号，如：

```
\begin{align}
f(x) =& \int (a + b)^2 dx \nonumber \\
     =& \int (a^2 + b^2 +  2ab) dx
\end{align}
```

结果如：

$$\begin{aligned} f(x) &= \int (a + b)^2 dx \\ &= \int (a^2 + b^2 + 2ab) dx \end{aligned} \tag{1.7}$$

不编号多行公式如:

```
\begin{align*}
  a + b =& c \\
  a^2 + b^2 =& c^2
\end{align*}

$$
f(x)>0, \quad \forall x \in R.
$$
```

结果如：

$$a + b = c$$
$$a^2 + b^2 = c^2$$

$$f(x) > 0, \quad \forall x \in R.$$

复杂公式和符号参见 `amslatex.pdf`，`symbols-a4.pdf`。

第二章　Emacs+AucTeX

对于熟悉 Emacs 的用户，用 Emacs+AucTeX 可以实现 LaTeX 的编写、编译、查看。其功能比 TeXWorks 和 WinEdt 更强，但是其使用不同于通常的 MS Windows 编辑程序，所以不建议一般 MS Windows 用户使用。新手建议使用 TeXWorks UTF-8 编码，XeLaTeX 编译。

Emacs+AucTeX 特点为:

- 语法高亮显示；
- 预览功能，即在编辑器内部直接显示公式编译结果和图形；
- 源文件和 DVI 结果正向和反向互查；
- 命令、环境、符号辅助输入；
- 主控文件和子文件支持；
- 公式、图标、文献等的索引支持；
- 编译菜单和快捷方式。

2.1　安装

先安装 MS Windows 用的 emacs-23.1.zip 文件，只要把它释放在合适的目录就可以了，比如安装在`c:\emacs` 中。

然后下载安装 auctex 软件包。只要把它释放到 emacs 安装目录就可以了，注意目录结构是以 emacs 安装目录为根的。

在 emacs 的`site-lisp` 目录中创建或修改`site-start.el` 文件，使其中包含行

```
;; for LaTeX Yap reverse search
(require 'gnuserv)
(gnuserv-start)
(setq gnuserv-frame (selected-frame))
```

```
(require 'tex-mik)
(require 'preview)
;; For LaTeX AucTeX mode accompanied by reftex mode.
(add-hook 'LaTeX-mode-hook 'turn-on-reftex)
(setq reftex-plug-into-AUCTeX t)


;; for selection using Ctrl-2
(global-set-key (quote [67108914]) (quote set-mark-command))

;; for Chinese environment
(set-language-environment 'Chinese-GB)
(set-keyboard-coding-system 'euc-cn)
(set-clipboard-coding-system 'euc-cn)
(set-terminal-coding-system 'euc-cn)
(set-buffer-file-coding-system 'euc-cn)
(set-selection-coding-system 'euc-cn)
(modify-coding-system-alist 'process "*" 'euc-cn)
(setq default-process-coding-system
            '(euc-cn . euc-cn))
(setq-default pathname-coding-system 'euc-cn)

;; for global font lock mode
(global-font-lock-mode t)

;; transient mark mode
(setq transient-mark-mode t)

;; show paren mode
(show-paren-mode t)
```

这样可以在每次启动 Emacs 时提供 LaTeX 编辑支持。Emacs 还会生成一个`C:\.emacs` 文件用来保存一些用户设置。

装完 Emacs 和 AucTeX 后再安装 CTEX 2.8。这样可以省去一些麻烦。运行 CTEX 中的 DVI 显示程序 Yap，选“View | Options”菜单再选“Inverse search”部分，如果其中 Program 有可选项“Gnu Emacs (Single instance)”，则选中这一项。这时对应的 Command 栏应显示

```
"C:\emacs\bin\gnuclientw.exe" -F +%l "%f"
```

这样可以实现用 Emacs+AucTeX 编译的 DVI 可以正反向位置对应的功能。

表 2.1: 光标移动及删除

操作	光标键	组合键	说明
右	右	C-f	**f**orward
左	左	C-b	**b**ackward
下	下	C-n	**n**ext
上	上	C-p	**p**revious
行首	Home	C-a	
行尾	End	C-e	**e**nd
下页	PageDown	C-v	
上页	PageUp	M-v	
文首	Ctrl-Home	M-<	
文尾	Ctrl-End	M->	
本行居中	无	C-l	
删除当前字符	Delete	C-d	
删除前一字符	Backspace	无	
从光标位置删除到行尾	无	C-k	**k**ill
取消刚才的删除	无	C-y	**y**ank
粘贴以前的删除	无	M-y 若干次	
恢复	无	C-x u	**u**ndo

如果 Yap 反查设置的对话框中不能出现 Emacs 的选项，可以增加新命令，在 Command 栏中输入以上命令，当然要写对路径。

2.2 Emacs 的基本编辑功能

在 Emacs 中用“Ctrl-x d”(按住 Ctrl 键按 x 键，放开 Ctrl 后按 d 键，此操作可简写为“C-x d”) 可以输入一个目录回车然后在窗口中管理这个目录，比如光标在某文件名上回车可以打开这个文件。命令“Ctrl-x Ctrl-f”可以打开一个旧文件或建立一个新文件。

Emacs 的编辑很有特点，表2.1列出了常用快捷键。注意用“C-x”表示按住 Ctrl 键再按 x 键，用“M-x”表示按住 Alt 键再按 x 键。Emacs 窗口最下面状态行下的地方叫做 mini-buffer，经常利用这里输入一些命令参数，如果要取消可以用“Ctrl-g”组合键。

注意“M-<”可能需要同时按住 Alt 和 Shift 键

复制、粘贴：在 Emacs 中选中一块文本，可以用鼠标拖动，也可以按

“Ctrl-2”组合键标记开始，把光标移动到另一位置标记结尾 (标记的文本可能不加亮显示，可在 Options 菜单中打开 Active Region Highlighting，这样在开始标记后可以加亮显示) 注意开始标记的命令本来是“Ctrl-@”，但是那样需要同时按 Ctrl 和 Shift，所以我们在`site-start.el` 文件中增加一行

```
(global-set-key (quote [67108914]) (quote set-mark-command))
```

就可以定义使用 Ctrl-2 来开始标记。标记后，可以用“Ctrl-w”命令剪切或用“Alt-w”命令复制，用“Ctrl-y”命令粘贴。用“Ctrl-x Ctrl-x”可以切换标记区的头尾。

在 Emacs 中保存了最近的若干次复制或剪切内容，先用“Ctrl-y”粘贴后可以用“Alt-y”替换成以前的内容。Emacs 中还记住了最近几次开始标记的地点，可以用“Ctrl-u Ctrl-2”命令返回到以前的标记点。

Emacs 是一个多窗口程序，可以同时编辑多个文件。Emacs 中的一个子窗口叫做一个 buffer(缓冲区)。状态行中显示了文件名，用鼠标左键或右键单击文件名可以切换要编辑的文件。

2.3 用 Emacs+AucTex 进行 LaTeX 编辑编译

如果打开或新建的文件名以“.tex”结尾则自动进入 LaTeX 编辑状态。增加了菜单 Preview、LaTeX 和 Command，Preview 菜单控制在 Emacs 内部的预览。LaTeX 菜单是辅助编辑命令，比如输入环境、命令、命令自动完成、查看编译错误等，Command 命令用于编译、浏览、转换 (PS, PDF) 等。

打开 LaTeX 文件后，如果要编译为 DVI 并可反查，不要选 Command 菜单中的“TeXing Options”的“PDF Mode”，选中 Command 菜单中的“TeXing Options”的“Source Specials”，用图标栏的 Run LaTeX 命令编译。如果有错误可以用“`C-c ``”键组合查看。用图标栏的 Run Viewer 命令可以打开 MikTeX 中的 Yap 程序对 DVI 进行浏览。Yap 中当前编辑位置用一个小圈显示出来。在 DVI 中移动到某一位置然后双击可以返回到 Emacs 中源文件的对应位置。

如果要编译为 PDF，则应该选中 Command 菜单中的“TeXing Options”的“PDF Mode”，然后用图标栏的 Run LaTeX 命令编译为 PDF，用图标栏的 Run Viewer 命令查看编译的 PDF。

在 Emacs 中用“Preview—Generate previews for document”可以为整个文章打开预览。用“Remov previews ...”关闭预览。可以预览选定部

分、所在环境等。

2.3.1 输入辅助

- “C-c C-m” 可以输入 LaTeX 命令 (macro)。在输入时可以用 TAB 键完成剩余部分。
- “C-c C-e” 可以输入 LaTeX 环境，在输入过程中还提示需要的选项。
- “C-c C-s” 可以输入章节命令。
- “M-RETURN” 可以输入列表环境的下一 item。
- “C-c C-f C-b” 产生\textbf{}。
- “C-c C-f C-e” 产生\emph{}。
- “C-c C-f C-t” 产生\texttt{}。
- “C-c {” 产生一对大括号。
- “C-c]” 关闭当前环境。
- “C-c C-q C-r” 排列当前 region 的格式 (indent)。“C-c C-q C-e” 排列当前环境。

2.3.2 注释

用 “C-c ;” 可以注释当前选定区域，用 “C-u - C-c ;” 可以取消当前区域的注释。

2.3.3 数学符号输入辅助

“C-c ~” 进入数学符号辅助输入状态，出现 Math 菜单可以输入常见数学符号，也可以以`为前缀加上一个单键输入数学符号。各希腊字母都可以用`加字母键输入。其他数学符号对应关系如下：

`< →≤ `> →≥ `I → ∞ `A → ∀ `E → ∃ `i →∈
`* → × `. → · `{ →⊂ `} →⊃ `[ →⊆ `] →⊇
`0 → ∅ `+ → ∩ `C-e → exp `c 字母键 → 花体字母

2.3.4 大文件辅助

AucTEX+Emacs 对分成几部分的 LaTeX 源文件支持很好，可以把文件头尾作为一个文件，如 mypap.tex，而其中的内容作为若干个文件，如 mypap1-inc.tex，mypap2-inc.tex 等。在主文件 (master file) mypap.tex 中内容的位置写命令 `\include{mypap1-inc}` 等可以包含入实际内容。一定注意`\include` 命令里面的文件名要省略.tex 后缀。

AucTeX 的好处在于：

- 即使内容处于不同文件中，编译查错仍可自动找到错误位置。
- Yap 的 DVI 反向查找可以正确找到子文件中相应位置。
- 用“C-c C-c”命令编译时即使当前文件是子文件实际编译的仍是主文件。这有可能需要自己在子文件末尾的地方加上类似如下的注释：

```
%%% Local Variables:
%%% mode: latex
%%% TeX-master: "mypap"
%%% End:
```

2.4 用 reftex 作引用

Emacs 对.tex 文件可以自动启用 reftex 支持，reftex 用来帮助查看文章的章节结果和辅助管理引用，如公式、图、表等。如果编辑在编辑.tex 文件时，按`C-=` 可以显示当前的目录结构，把光标移动到目录中任一位置可以移动到源文件中该章节位置。在目录 buffer 中用 l 命令可以显示当前定义的引用，r 命令可以重新扫描目录结构。在源文件中用`C-)` 可以提示输入引用 (`\ref{...}`)，选 e 可以提示公式的引用，在目录中选取一个引用，选哪个公式的引用时该公式的源程序会显示在状态行。

第三章　LaTeX 提高

3.1　编译结果

LaTeX 编译的结果有:

- PDF, 是一种不依赖于操作系统和语言环境的文档格式，一般不可编辑，在任何操作系统和语言环境都可以不变样显示和打印。
- Postscript(PS)，和 PDF 类似，因为现在 PDF 的浏览程序更好用所以 PS 格式逐渐被 PDF 格式取代。
- DVI 格式，是 TeX 特有的不依赖于显示系统的输出格式，编译为 DVI 比编译为 PDF 要快一些，能与源文件正向、反向互查位置，所以在写草稿时可以用。最终稿一般输出为 PDF。
- HTML 格式，这是网上的文章用的格式，显示于网络浏览器中。

编辑器 TeXWorks 编译产生 PDF 文件, 可以在源文件和结果之间互相查找。可以用 pdflatex 或 xelatex 编译。

WinEdt 提供了快捷图标可以:

- 编译为 DVI, 可以在源文件和结果之间互相查找；
- 直接编译为 PDF(用 PDFLaTeX)；
- DVI 转换为 PS；
- DVI 转换为 PDF；
- PS 转换为 PDF。

Emacs+AucTeX 提供了菜单和快捷图标可以:

- 选择编译为 DVI 还是 PDF；
- 编译，根据选择结果编译成 DVI 或 PDF；

- 清除辅助文件；
- 用“Command—Other”菜单，可以运行 `dvips` 文件名命令，把 DVI 转换为 PS; 运行 `dvipdfm` 文件名命令，把 DVI 转换为 PDF; 把 LaTeX 源文件转换为 HTML 文件。

3.2 带书签的 PDF 文件

用 LaTeX 的 Hyperref 包可以生成带书签和超链接的 PDF 文件。这样的 PDF 文件在浏览时可以在左窗格显示文件的目录结构，单击某一章节可以迅速跳到该处。文中的交叉引用如图表、章节、文献等都可以通过链接直接跳到定义处。

使用 Hyperref 包等功能十分容易，只要在引言 (preamble) 部分加上

```
\usepackage[pdftex,colorlinks=true]{hyperref}
```

则用 PDFLaTeX 生成的 PDF 文件自动带有书签和超链接。但是，如果我们使用 UTF-8 中文编码用 XeLaTeX 编译，则应去掉 pdftex 选项。

Hyperref 包提供的 `\hypertarget{`*name*`}{`*text*`}` 命令可以定义一个文内超链接，可以用 `\hyperlink{`*name*`}{`*text*`}` 来引用。

3.3 版式设置

用

```
\pagestyle{plain}
```

产生只有页号的页。用

```
\pagestyle{headings}
```

产生有页眉且页码在页眉的页面。

为了设置版心宽度和左右边空，用如

```
\setlength{\textwidth}{12.1cm}
\setlength{\oddsidemargin}{2.46cm}
\setlength{\evensidemargin}{1.36cm}
```

设置了版心宽为 12.1 厘米，纸张内侧边界为 1in+2.46cm=5cm，外侧边界为 1in+1.36cm=3.9cm，总共宽 21.0cm(A4 纸宽度)。用这样的命令设置边空要加上缺省的 1 英寸 (2.54 厘米)。A4 纸高度为 29.7 厘米，可以用 \textheight 和 \topmargin 设置版心高度，注意上边空也要加上缺省的 1 英寸。

如果要设置自己定义的页眉和页脚，可以使用 fancyhdr 包。

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

引入该包并声明使用其版式。

3.4 大文件处理

有一些技巧可以处理大的文件。

- 可以把文件头、尾写成一个主控文档 main.tex，然后在文档内部使用\include{文件名} 的方法插入真正的内容文件。注意文件名不要包括.tex 扩展名。\include 会在插入内容前后换页。

- 如果有多个文件插入，还可以在前言 (preamble) 部分加一个 \includeonly{文件清单} 命令要求只插入列出的部分文件。如果没有\includeonly 命令则正文中所有用\include 命令插入的文件都得到处理，如果有\includeonly 命令则只处理指定的文件，但是页码、索引等仍可以用到全文的信息。AucTeX+Emacs 可以很好地支持多文件源程序，可以正确显示错误位置、Yap DVI 反查。

- 主文件可以有几个，共享相同的内容子文件，可以适应不同的设置或输出需要。这样把内容和头尾分开有明显的好处：内容只需要一份，可以与几个不同的主控文件配合，比如生成 DVI 用的主控文件，生成 PDF 用的主控文件，生成课堂演示 PDF 的主控文件，CTeX 用的主控文件。

- 为了临时注释掉一部分内容，可以用 \iffalse 开头用 \fi 结尾把该部分包裹起来。为了容易找到这些注释标志可以在其附近加上若干个分号注释符。

3.5 插入图形

LaTeX 插图主要用 graphicx 包，它可以用`\includegraphics` 命令插入一个图形。可以直接插入到段落中等同于一个字符或段内公式，也可以放在 LaTeX 的 figure 环境内作为浮动图形。放在 figure 环境内时一般用 center 环境把图形居中。

编译成 DVI 时可以插入的图形主要是 EPS 图形。一般我们生成的单页的 PS 图形都可以看成 EPS 图形，对某些不能正常计算边框的 PS 图形可以在新版的 gsview 中打开然后使用其中的 “PS to EPS” 菜单。要把其它格式的图形转换为 EPS 图形有多种方法：可以用 GIMP、Photoshop 等图象处理图形进行转换，也可以安装一个支持 PS 的打印机驱动程序利用其打印到文件的功能把图形打印成 PS 文件，等等。

用 PDFLaTeX 和 XeLaTeX 编译时可以支持 PDF、PNG、JPEG、GIF 等格式的图形。为了把 PS 或 EPS 文件转换为 PDF 可以用新版的 gsview 中的转换功能。

注意我们插图比较理想的格式是 PDF 和 EPS，因为这两种格式支持矢量图形，在放大时不会失真。如果原来的图形就是点阵格式如 PNG、JPEG、GIF 就不必转换为 PDF 了。

`\includegraphics` 命令的可选参数主要是 width= 和 height =，如果希望保持原图的高宽比就只要指定 width 和 height 之一。注意比较大的插图一定不要设定超过行宽的宽度，否则会产生很多多余的空白。

如果我们有大量的图，不希望和 LaTeX 源文件放在同一目录，可以把图放在一个 figs 子目录，假设还有一个 grs 子目录，只要在主控文件前言 (preamble) 部分加上

```
\graphicspath{{figs/}{grs/}}
```

就可以自动从这两个目录中查找图形来插入。

图3.1是插入单个浮动图形的例子。用到的命令如下：

```
\begin{figure}[htbp]
\centering
\includegraphics[width=\textwidth]{img01}
\caption{插入单个浮动图的例子 \label{fig:figfloat1}}
\end{figure}
```

浮动图形环境的选项 h 表示当前位置，t 为 top，b 为 bottom，p 为单独只有图形的一页。注意`\label` 放在`\caption` 内部比较保险。

如果浮动图形位置不满意可以把插图命令在正文中前后调整。

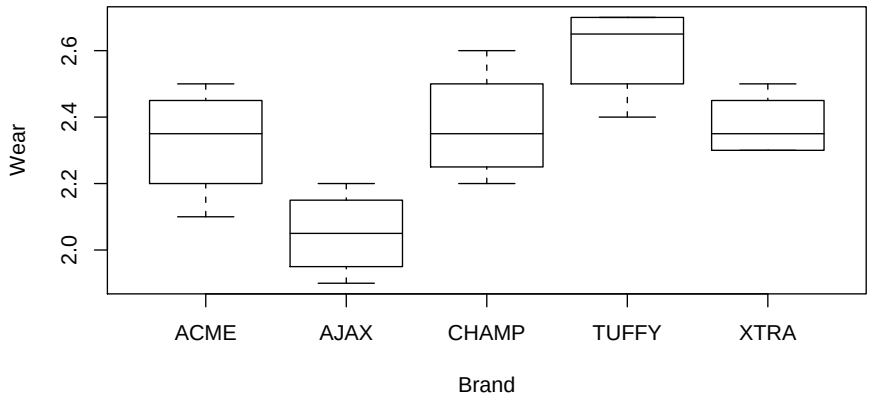


图 3.1: 插入单个浮动图的例子

图3.2为边上插入图形的一个例子。用了特殊的 picins 包，这个包主要提供了不是浮动而是在当前段左边或右边插图的能力，图形还可以有各种边框。插图命令为\parpic，可选参数[r]说明放在段右侧，后面的必要参数为实际的插图，我们用自定义的\img 命令插入了图形。这样插图的说明标题要用 \piccaption 放在实际插图命令\parpic 前面才行。这个例子中的\pichskip 指定图横向离开文本段多宽，不是必须的。源程序如下：

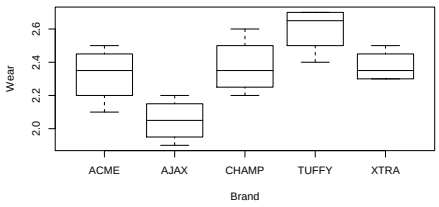


图 3.2: 用 picins 包插入的靠边图

```
\piccaption{用 picins 包插入的靠边图 \label{fig:parpic}}
\parpic[r]{
\pichskip{6em}
\includegraphics[width=0.5\textwidth]{img01}}
```

这个插图方法还不太完善，有时得到不好的结果。

另外一种插入边图的办法是使用 floatflt 宏包，有时会覆盖在后面段落，不理想。

对于比较小的图形我们希望把两个并排放在一起浮动，但是有各自的图形标题，可以用下例的方法，结果见图3.3和图3.4：

```
\begin{figure}[htbp]
  \begin{minipage}[t]{0.48\linewidth}
    \centering
    \includegraphics[width=\textwidth]{img01}
    \caption{这是第一个图 \label{fig:float2-1}}
  \end{minipage}
  \hfill
  \begin{minipage}[t]{0.48\linewidth}
    \centering
    \includegraphics[width=\textwidth]{img02}
    \caption{这是第二个图 \label{fig:float2-2}}
  \end{minipage}
\end{figure}
```

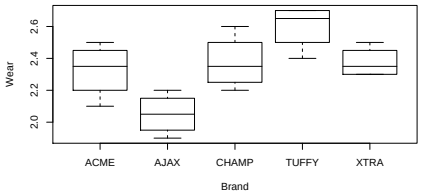


图 3.3: 这是第一个图

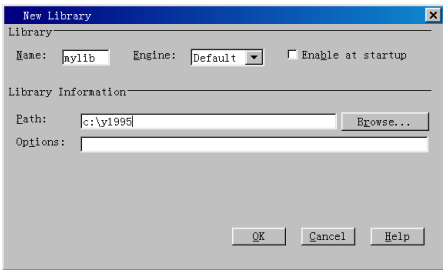


图 3.4: 这是第二个图

在 R 中可以直接生成 PDF 图形, 如:

```
pdf("hor.pdf",
    width=20/2.54, height=16/2.54, horizontal=FALSE)
```

这样得到的图在插入 LaTeX 时不需要旋转。

插入的浮动图形太多时，可能会出错（Too many floats）。解决方法可以在适当地方插入`\clearpage` 命令，或引入 placeins 包，然后在适当地方使用`\FloatBarrier` 命令。

3.6 源程序排版

源程序排版使用 fancyvrb 宏包。用如下命令可以设置一些统一的选项：

```
\RecustomVerbatimEnvironment{Verbatim}{Verbatim}{%
frame=single,framesep=5pt,xleftmargin=1cm,fontsize=\normalsize}
```

这样要求外边框 (frame)，框内左边空 5pt，左外边界 1cm，字体用正常大小。Verbatim 环境也可以单独加选项, 如 \begin{Verbatim}[fontsize=\small]。

可以用`\VerbatimImput{file.ext}` 导入一个外部文件内容，显示为原样显示。此命令可以加选项 [frame=single] 要求画边框。

为了排一些程序模板，可以用 tabular 环境对齐。如下例:

```
\begin{tabular}{lll}
&&\\
\multicolumn{3}{l}{SELECT (\emph{选择表达式});} \\
\quad & WHEN(\emph{值列表})  &  \emph{语句};    \\
\quad & WHEN(\emph{值列表})  &  \emph{语句};    \\
\quad & …… &                                    \\
\quad & OTHERWISE            &  \emph{语句};    \\
\multicolumn{3}{l}{END;}                          \\
&&
\end{tabular}
```

结果如下 (非浮动)：

SELECT (选择表达式);
 WHEN(值列表) 语句;
 WHEN(值列表) 语句;
 ……
 OTHERWISE 语句;
END;

其中在 tabular 环境选项的两列之间用了`@{ }` 来取消表格环境原来比较宽的列间距。表格前后各多加了一个空行以区分上下文。不想对齐的部分用 mulitcolumn 命令合并即可。

还可以使用 tabbing 环境进行对齐。如

```
\begin{minipage}{0.8\textwidth}
\begin{tabbing}
\qquad \= if \= \kill
```

```
\> if(条件) \{  \\

\>\> 表达式组 \ldots \ldots \\

\> \} else \{ \\

\>\> 表达式组 \ldots \ldots \\

\> \} \\
\end{tabbing}
\end{minipage}
```

结果为：

if(条件) {
 表达式组……
} else {
 表达式组……
}

第四章　幻灯片演示制作

现在人们一般用 Powerpoint 作演示，但是如果公式比较多就不方便。另外，如果你原来的文章是用 LaTeX 打的那么用 LaTeX 做幻灯片就是最合理的选择。用 beamer 宏包或 PDFSlides 宏包可以制作课堂演示用的幻灯片。Beamer 制作的幻灯片比较漂亮。

4.1　用 Beamer 制作幻灯片

beamer 是一个制作 PDF 幻灯片的包。其优点包括:

- 自动生成导航菜单；
- 支持自动的分步显示 (列表环境) 和定制的分步显示；
- 带有不同的色彩、样式；
- 除支持生成屏幕显示的幻灯片 PDF 外，用相同的输入可以得到普通打印版 PDF，且可以选择那些内容进入幻灯片，那些内容进入打印版；
- 支持 CJK 中文和 UTF-8 中文。

4.1.1　基本的 beamer 幻灯片

安装好 beamer 包后，用 beamer 制作幻灯片 PDF 的基本框架为 (文件 slides.tex, 采用 UTF-8 编码，用 XeLaTeX 编译):

```
% !TEX encoding = utf8
% !TEX program = xelatex

%%% 关于 LaTeX 的入门介绍。

\documentclass{beamer}
```

```
\usepackage[noindent]{ctex}

\usepackage{amsmath}                    % 数学符号与公式

% 设置 beamer 中公式字体
\usefonttheme{professionalfonts}
% 设置中文字体
\setCJKmainfont[BoldFont={黑体},ItalicFont={楷体}]{新宋体}

\hypersetup{
  pdfauthor = {Author Name},
  pdftitle = {Your Title},
  colorlinks,
  bookmarks=false,
  pdfstartview={Fit},
  pdfpagelayout={SinglePage},
  pdfpagemode={None}
} % hyperref 包的设置。

\usetheme[hideothersubsections]{Goettingen} % 菜单在右，浅蓝白色
\setbeamercovered{transparent}

\begin{document}

\title{演讲题目}
\subtitle{
小标题一 \\
小标题二
}
\auther{作者名 \\
单位名}
\date{日期}
\begin{frame}
  \maketitle
\end{frame}

\include{mypres}

\end{document}
```

其中，\usetheme 指定一种样式，包括 Antibes, Bergen, Berkeley, Berlin, Boadilla, Copenhagen, Darmstadt, Dresden, Frankfurt, Goettin-

gen, Hannover, Ilmenau, JuanLesPins, Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester, Singapore, Szeged, Warsaw 等，比如 Warsaw 生成在上面的菜单，Berkeley 生成在左面的菜单，Goettingen 生成在右面的菜单，Berlin 生成可显示每小节页面数的菜单，Rochester 生成没有菜单的样式。不同的样式有不同的颜色、字体、导航。

为了在开头加入目录，并在每节前面显示本节加亮显示而其它节透明显示的目录，在标题页后加入

```
\begin{frame}[shrink=10]
\frametitle{本章目录}
\tableofcontents[hideallsubsections]
\end{frame}

\AtBeginSection[]
{
  \begin{frame}[shrink=10]
    \frametitle{本节目录}
    \tableofcontents[currentsection,hideothersubsections]
  \end{frame}
}
```

beamer 中一面幻灯片用一个 frame 环境实现。在一个 frame 中，可以用 \pause 命令定制需要暂停显示的地方。对于 itemize 环境，还可以自动每个条目暂停，如:

```
\begin{frame}
  \frametitle{内容标题}
  \begin{itemize}[<+->]
  \item 条目一
  \item 条目二
  \end{itemize}
\end{frame}
```

其中 \frametitle 加入屏幕标题。

虽然 beamer 支持把长的 frame 自动分成多个屏幕显示，但是拆分会造成一些问题，最好手工拆分长的 frame。

4.1.2 编译速度

编译一个长的演讲稿每次比较慢。为此，建议使用 TeXWorks 来编辑和编译，可以准备把主控文件 slides.tex 复制为 drafts.tex, 引入类 beamer 时加入 draft 选项，如

```
\documentclass[CJK,draft]{beamer}
```

然后，对正在编写的屏幕 (frame) 加 [label=current] 选项，如`\begin{frame}[label=current]`, 则对该屏幕定义了一个标签 current，然后在序言中用

```
\includeonlyframes{current}
```

只编译此屏幕。当一个屏幕调试成功后再注释此句以编译全文档。

4.1.3 同时生成幻灯片和打印版本

只要我们把主要内容放在单独的文件内，为幻灯片版本和可打印版本单独写主控文件就可以。beamer 对可打印版的生成支持很好，在打印版中 \frame 等命令自动被转换含义。

打印版本的主要框架为:

```
\documentclass[a4paper]{ctexbook}
\usepackage{beamerarticle}
\setCJKmainfont[BoldFont={黑体},ItalicFont={楷体}]{新宋体}
\usepackage[colorlinks=true]{hyperref}

\begin{document}
\include{mypres}
\end{document}
```

也可以用其它文档模板。beamerarticle 包的调入解决了正文中 frame, pause 等命令的解释问题。

如果有些内容只希望出现在幻灯片中，可以用

```
\only<presentation>{
只出现在幻灯片中。
}
```

如果希望只出现在可打印版本中只要用 `\only<article>`。

4.2 用 PDFSlide 制作幻灯片

LaTeX 另一种制作幻灯的方法是用 pdfslide 宏包。只要在文件头用`\usepackage[ams]{pdfslide}` 调用这个包，用 PDFLaTeX 编译就可以生成适合屏幕演示的 PDF 文件。源文件最好使用 article 类。生成的演示用回车翻页，带有导航图标。因为演示是基于屏幕的，在不同屏之间可以用`\clearpage` 命令换页。

文章中可以用`\color{`颜色名`}` 改变字体颜色。用`\overlay{metablue.pdf}` 命令可以给屏幕加一个背景，这个背景可以用自己的 PDF 格式或 JPEG 格式图形代替。用`\pagedissolve{Wipe /D 1}` 命令可以定义各页之间的转场效果。其中`/D 1` 表示延迟 1 秒。

可用的转场效果为：

- `\pagedissolve{Replace}` 无转场效果。这是缺省取值。
- `\pagedissolve{Split /D 1 /Dm /V /M /I}` 垂直线向内拉幕
- `\pagedissolve{Split /D 1 /Dm /V /M /O}` 垂直线向外拉幕
- `\pagedissolve{Split /D 1 /Dm /V /M /I}` 水平线向内拉幕
- `\pagedissolve{Split /D 1 /Dm /H /M /O}` 水平线向外拉幕
- `\pagedissolve{Blinds /D 1 /Dm /H}` 多垂直线拉幕
- `\pagedissolve{Blinds /D 1 /Dm /H}` 多水平线拉幕
- `\pagedissolve{Box /D 1 /M /I}` 向内盒子
- `\pagedissolve{Box /D 1 /M /O}` 向外盒子
- `\pagedissolve{Wipe /D 1}`，`\pagedissolve{Wipe /D 1 /Di 90}` 向上推拉
- `\pagedissolve{Wipe /D 1 /Di 180}` 向左推拉
- `\pagedissolve{Wipe /D 1 /Di 270}` 向下推拉
- `\pagedissolve{Dissolve /D 1}` 渐化
- `\pagedissolve{Glitter /D 1 /Di 0}` 推拉渐化

下面是背景和转场效果的一个实例：

```
\overlay{bg151.jpg}
\pagedissolve{Wipe /D 0.5}  % 向右推拉
```

其中 bg151.jpg 是我自己在图形程序中作的一个背景图片。注意背景不要喧宾夺主，否则对教学效果会起反面作用。

在文件头加上如下定义可以得到较悦目的项目列表：

```
\renewcommand{\labelitemi}{\textcolor{red}{$\bullet$}}
\renewcommand{\labelitemii}{\textcolor{yellow}{$\star$}}
\renewcommand{\labelitemiii}{\textcolor{magenta}{$\ast$}}
\renewcommand{\labelitemiv}{\textcolor{cyan}{$\circ$}}
```

如果必要的话还可以下载安装 PPower4 软件以生成逐步显现的页。

用 PDFSlide 包作的课堂演示文件没有目录导航功能。我们可以用 Hyperef 包提供的两个命令自己建立文内的超链接：\hypertarget{链接名}{显示内容} 可以定义一个链接目的地，用“链接名”来访问，为了显示这个超链接，用\hyperlink{链接名}{显示内容} 显示。

演示文件是以屏幕为单位的，需要自己分页。在生成课堂演示的主控文件中我们定义了：

```
\providecommand{\cls}{\clearpage}
```

而在另外一个主控文件中我们则定义

```
\providecommand{\cls}{\vskip 1cm}
```

这个主控文件可以生成课下参考用的讲义。

因为 PDFLaTeX 编译比较慢所以我们先编译成 DVI 成功后再编译成 PDF，这样需要一个单独的生成 DVI 的主控文件。为了和结果屏幕大小一致，我们定义文本高度：

```
\setlength{\textheight}{10.6cm}
```

4.2.1 PPower4 软件

为了实现逐步显示一页的各行的效果需要安装一个 PPower4 软件包。这是一个 Java 程序，所以需要先安装一个 Java 运行环境，如 Java JRE 1.2。下载 PPower4 后把其中的 pp4sty.zip

展开到 C:\CTEX\MikTeX\LocalTeXMF\tex\latex\ppower4 中，把其中的 pp4p.jar 复制到上述目录中，然后生成如下的批命令 PPower4.bat:

```
set basedir="C:\Program Files\JavaSoft\jre\1.2\lib"
set javacommand="C:\Program Files\JavaSoft\jre\1.2\bin\java"
%javacommand% -jar C:\CTEX\MiKTeX\LocalTeXMF\tex\latex\
                        ppower4\pp4p.jar %1 %2 %3 %4 %5
```

把它放在 C:\CTEX\MiKTeX\TeXMF\miktex\bin 中。注意批命令中需要代以你自己的 Java 运行环境的路径和程序名。展开 pp4sty.zip 中的文件到 CTEX 目录中后要更新 MikTeX 的文件名数据库。

PPower4 主要是提供了一个 pause.sty 包和\pause 命令。为使用其功能需要在 LaTeX 文件头引入 pause 包。在需要等待回车继续显示的地方加上\pause 命令。然后对 LaTeX 源文件 (如 test.tex) 用 PDFLATEX 命令编译，然后运行 PPower4:

```
ppower4 test.pdf test-scr.pdf
```

则 test-scr.pdf 为生成的结果。

PPower4 除了提供分步显示支持还提供了单色、渐变色背景，转场动画效果等。

第五章　其他技巧

5.1　笔记

5.1.1　一些自定义命令

```
\providecommand{\bs}{\boldsymbol}
\newcommand{\defeq}{\stackrel{\triangle}{=}}
\newcommand{\cfracop}[1]{\genfrac{}{}{0pt}{}{}{#1}}
\newcommand{\limdist}{\stackrel{\mbox{d}}{\longrightarrow}}
\newcommand{\Expect}{\ensuremath{\mathrm{E}}}
\newcommand{\expect}{\ensuremath{\mathrm{E}}}
\newcommand{\tr}{\ensuremath{\mathrm{tr}}}
\providecommand{\det}{\ensuremath{\mathrm{det}}}
\newcommand{\Var}{\ensuremath{\mathrm{Var}}}
\newcommand{\VVar}{\ensuremath{\mathrm{Var}}}
\newcommand{\Cov}{\ensuremath{\mathrm{Cov}}}
\newcommand{\SE}{\ensuremath{\mathrm{SE}}}
\providecommand{\Prob}{\ensuremath{\mathrm{Pr}}}
\newcommand{\diag}{\ensuremath{\mathrm{diag}}}
\newcommand{\ones}{\ensuremath{\mathds{1}}}
\newcommand{\EbNo}{\ensuremath{\mathrm{E_b/No}}}
\newcommand{\WN}{\ensuremath{\mathrm{WN}}}
\newcommand{\inprod}[2]{\langle #1, #2\rangle}
\providecommand{\Rbb}{\mathbb R}
\providecommand{\Nbb}{\mathbb N}
\providecommand{\Nbbp}{\mathbb N_+}
\providecommand{\Zbb}{\mathbb Z}
\providecommand{\bm}[1]{\boldsymbol{#1}}
\providecommand{\BB}{\mathscr B}
\providecommand{\FF}{\mathscr F}
\providecommand{\argmin}{\mathop{\mathrm{arg\,min}}}
\providecommand{\argmax}{\mathop{\mathrm{arg\,max}}}
\newcommand{\veps}{\varepsilon}
```

```
\newcommand{\vv}{\varepsilon}
\newcommand{\ga}{\gamma}
\newcommand{\sig}{\sigma}
\providecommand{\lspan}{\mathop{\overline{\text{sp}}}}
```

5.2 公式和符号问题

在网上搜索一个 symbols-a4.pdf 文件，这个文件中罗列了很多符号的用法。比如，空心的阿拉伯数字如何打，黑体希腊字母用什么样的命令。

一些宏包提供了更多的符号。amsmath 提供额外的数学公式能力，如 align 等。amsbsy 提供增强的 \boldsymbol 命令，可以排版粗体的希腊字母等粗体符号。mathrsfs 提供了花体的 B, F 等。gensymb 提供了 \celsius, \degree 等符号。mathrsfs 提供了 math script 字体。dsfont 提供黑板手写粗体符号，包括数字 1,2 等。

5.2.1 特定问题

如何定义公式中一些类似于 ln 的命令

比如，Cov 在公式中可以用

```
\providecommand{\Cov}{\ensuremath{\mathrm{Cov}}}
```

定义后用`$\Cov$` 调用。

为了生成如 $\lim_{n\to\infty}$ 这样可以带上下标的命令，可以定义如

```
\providecommand{\argmin}{\mathop{\mathrm{arg\,min}}}
```

如

```
$$
\argmin_{x \in A} f(x)
$$
```

生成

$$\arg\min_{x\in A} f(x)$$

如何生成公式中的粗体字

引入 amsmath 包后，用 \boldsymbol x 产生 $\boldsymbol{x}$，用 \boldsymbol \alpha 产生 $\boldsymbol{\alpha}$。

如何生成如 $\mathscr{F}$ 这样的花体字

需要引入 mathrsfs 包然后用如`$\mathscr F$`。

如何生成如 $\mathbb{R}$ 这样的空心字

只要引入了 amsfonts 包或 amssymb 包就可以用 `$\mathbb R$` 生成。

如何生成如 $\mathbb{1}$ 这样的空心数字

可以在引入 dsfont 包:

```
\usepackage[sans]{dsfont}
```

后用`$\mathds 1$` 实现。

5.3 其他问题

5.3.1 关于英文系统和中文系统的兼容性问题

我们常用的 CTEX 安装了一些关于汉字系统的设置，在少数情况下编译一些比较复杂的纯英文文件会因为不兼容而出错。这时我们可以采用更改 MikTeX 的目录次序的办法：在安装好的 MikTeX 程序组中找到“MikTeX Options”程序，启动后，有一个“File name database—Refresh Now”选项，这个选项是在安装了新的包后应该运行的。点这个程序的 Roots 选项卡，其中有两个目录，`\CTeX\localtexmf` 和`\CTeX\texmf`，前者保存了中文系统和附加的包，后者为 MikTeX 安装的主要 TeX 系统。这两个目录有次序，当 localtexmf 在前时中文系统为优先考虑，当 texmf 在前时英文系统为有限考虑。所以为了编译不兼容的英文文件可以修改这个次序，工作完成后再改回来。

5.3.2 一些小的问题

如何临时增加一页高度

有时 LaTeX 会生成一页只有一两行的几乎空的页，有时插入一个较高的图表导致前面出现一个空页。为避免这种情况，可以在内容的源代码中加入

```
\enlargethispage{2\baselineskip}
```

就可以放大内容的一页使其能容纳较多内容而不至于换页。

如何修改 enumerate 环境的序号

为了自己设计 enumrate 环境的需要，可以在环境内部开始的地方重新定义\labelenumi, \labelenumii 等命令。比如

```
\renewcommand{\labelenumi}{\textbf{\arabic{enumi}.}}
\renewcommand{\labelenumii}{(\alph{enumi})\quad}
```

等。也可以在\item 命令的可选参数中直接给出序号，如\item[一、]。

如何避免浮动图表不必要地占据整页

LaTeX 关于图表等浮动体的放置可以由 htbp 等建议，在\begin{figure}[htbp] 中的 h 表示建议优先放在当前位置，其次放在页首，再其次放在页尾，实在不行允许单独放在一页。在具体实施时，图表的放置位置有一些默认的限制，比如图文混排时如果文字部分过少则把图单独放在一页。这些限制可以用如下方法进行调整:

```
\renewcommand{\textfraction}{0.10}
\renewcommand{\topfraction}{0.90}
\renewcommand{\bottomfraction}{0.70}
\renewcommand{\floatpagefraction}{0.90}
```

其中 textfracton 的设置把混排时最少文本降低到 10%，topfraction 的设置允许页首的图占整页的 90%(图高超过这个比例时则只能单独放在一页)，bottomfraction 的设置允许页尾的图占整也的 70% (图高超过这个比例时不允许放在也尾)，floatpagefraction 的设置要求图高大于 90% 时才单独放在一页。

插入过多图形或源程序出错

当源文件中有过多的原样文本 (verbatim) 时可能会发生 “no room for new \box” 问题。可以引入 etex 宏包，如

```
\usepackage{etex}
\reserveinserts{30}
```

如何制作横向表格

太宽的表格可以横过来排印，这样的表格要单独占一页。LaTeX 的 lscape 包可以提供横向排版功能，lscape 提供了一个 landscape 环境，在此

环境内排版是横向的。